

Supplementary of “Rethinking Optical Flow from Geometric Matching Consistent Perspective”

Qiaole Dong*, Chenjie Cao*, Yanwei Fu†
 School of Data Science, Fudan University

{qldong18, 20110980001, yanweifu}@fudan.edu.cn

1. Details of QuadTree Attention

In the Feature Matching Extractor (FME), we employ the QuadTree attention [5] to enhance the feature. Specifically, given the image feature F_1, F_2 from ResNet-16, 8 stacked QuadTree attention blocks (4 self-attention and 4 cross-attention blocks) are incorporated into FME to enhance F_1 and F_2 . We first linear project F_1 and F_2 to query Q , key K , and value V . Take cross-attention as an example:

$$Q = W_q F_1, \quad (1)$$

$$K = W_k F_2, \quad (2)$$

$$V = W_v F_2, \quad (3)$$

where W_q, W_k, W_v are learnable parameters. We then construct 3-level pyramids for query Q , key K , and value V by average pooling. After computing attention scores at the coarse level:

$$A = \text{Softmax}\left(\frac{QK^T}{\sqrt{C}}\right), \quad (4)$$

we select the top k key tokens with the highest attention scores for each query token. At the finer level, query sub-tokens only need to be evaluated with those key sub-tokens that correspond to one of the selected k key tokens at the coarse level. This process is repeated until reaches the finest level. We finally weighted average all selected value tokens at all levels through learnable weight and attention scores. And k is set to 16 for the coarsest level, and 8 for the remaining levels.

2. Tile Technique

As KITTI owns a much smaller aspect ratio, we use the tile technique [2, 3]. Specifically, given a test image with size (H_{test}, W_{test}) , we split it into several patches according to training image size (H_{train}, W_{train}) . For example, it results in two patches starting at $(0, 0)$ and $(0, W_{test} - W_{train})$ if $H_{test} \leq H_{train}$; and four patches starting

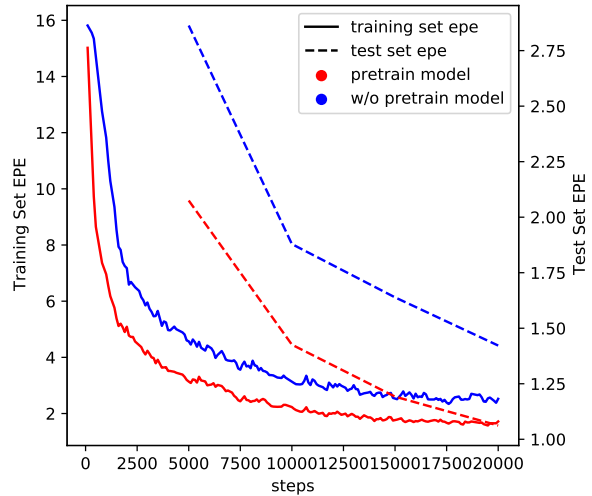


Figure 1. EPE graph at early iterations on C+T pre-training.

at $(0, 0)$, $(H_{test} - H_{train}, 0)$, $(H_{test} - H_{train}, W_{test} - W_{train})$, and $(0, W_{test} - W_{train})$ otherwise. For pixels covered by several patches, we weighted average the flows from these patches and get the final results. The weight is computed from the pixel’s normalized distances $d_{u,v}$ to the corresponding patch center:

$$d_{u,v} = \|(u/H_{train} - 0.5, v/W_{train} - 0.5)\|_2, \quad (5)$$

where (u, v) is the pixel’s 2D index within each patch. And we use the Gaussian probability density function to get the final weight for each patch:

$$w_{u,v} = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{d_{u,v}^2}{2\sigma^2}\right), \quad (6)$$

where $\sigma = 0.05$.

3. How much does the Megadepth pretraining provide a good starting point?

We further provide an EPE graph at early iterations on C+T pre-training, for the Megadepth-trained model and a

*Equal contributions.

†Corresponding author.

model from the scratch. Pretrain indeed provides a much better starting point and converges to lower error on training/test set as shown in Fig. 1.

Besides, as we only pretrain feature encoder by GIM, flow decoder is still learned from scratch on flow data. Directly finetuning model on C+T+S+K+H can result in poor performance (1.63 and 2.77 on Clean and Final of Sintel test set respectively) and serious grid artifacts around motion boundary. So the synthetic dataset pretraining (C+T) is still necessary for our method.

4. More Qualitative Comparison

More qualitative results on Sintel test set and KITTI set are compared between our MatchFlow(G) and GMA [4] are given in Fig. 2 and Fig. 3. As these samples from Sintel test set have no ground-truth optical flows, we can not give the AEPE and replace the ground-truth flows with matching frames in the second column in Fig. 2. We highlight the areas where our MatchFlow(G) beats GMA [4]. Please zoom in for more details.

In addition, we provide qualitative comparison with GMA [4] on HD video from DAVIS [1] test set. We test models on 1080p (1088x1920) resolution video and set the GRU iterations to 12 for both models. We do not use tile technique [3] here. Both models are trained on Sintel. Fig. 4 shows that our model exhibits clearer details (first and third rows) and performs better on textureless regions (second row). Please zoom in for more details.

5. Method of Correlation Volume Visualization

We visualize the correlation volume following GM-FlowNet [6]. Specifically, given 4D correlation volume: $C \in \mathbb{R}^{H \times W \times H \times W}$, where i, j indicate the index of feature map F_1 and F_2 ; H, W indicate 1/8 height and width of the input image, we extract the local correlation map F_i for point $i = (u, v)$ around the ground-truth optical flow $f_{gt} = (f_{gt}^1, f_{gt}^2)$ as follows:

$$F_i = C(i, (u + f_{gt}^1 + x, v + f_{gt}^2 + y)) \in \mathbb{R}^{1 \times 1 \times 11 \times 11}, \\ -5 \leq x \leq 5, -5 \leq y \leq 5. \quad (7)$$

As H, W indicate 1/8 height and width of the input image, a local 11×11 window in C corresponds to 88×88 local window in input image. We then normalize the local correlation map by *Softmax*:

$$\hat{F}_i = \text{Softmax}(F_i). \quad (8)$$

Finally, we average \hat{F}_i on all points within different region on 100 Sintel final pass images. The results are shown in Fig. 4 of main paper.

6. Screenshots of Sintel and KITTI Results

We provide anonymous screenshots of Sintel and KITTI results on the test sever in Fig. 5 and Fig. 6. Our method ranks first on Sintel Clean pass and sceond on Sintel Final pass among all published approaches. Besides, we also achieve great performance improvement on KITTI test set. These results signifies the effectiveness of our approach.

References

- [1] Sergi Caelles, Jordi Pont-Tuset, Federico Perazzi, Alberto Montes, Kevis-Kokitsi Maninis, and Luc Van Gool. The 2019 davis challenge on vos: Unsupervised multi-object segmentation. *arXiv*, 2019. 2, 4
- [2] Zhaoyang Huang, Xiaoyu Shi, Chao Zhang, Qiang Wang, Ka Chun Cheung, Hongwei Qin, Jifeng Dai, and Hongsheng Li. Flowformer: A transformer architecture for optical flow. *arXiv preprint arXiv:2203.16194*, 2022. 1
- [3] Andrew Jaegle, Sebastian Borgeaud, Jean-Baptiste Alayrac, Carl Doersch, Catalin Ionescu, David Ding, Skanda Koppula, Daniel Zoran, Andrew Brock, Evan Shelhamer, et al. Perceiver io: A general architecture for structured inputs & outputs. *arXiv preprint arXiv:2107.14795*, 2021. 1, 2
- [4] Shihao Jiang, Dylan Campbell, Yao Lu, Hongdong Li, and Richard Hartley. Learning to estimate hidden motions with global motion aggregation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9772–9781, 2021. 2
- [5] Shitao Tang, Jiahui Zhang, Siyu Zhu, and Ping Tan. Quadtree attention for vision transformers. *arXiv preprint arXiv:2201.02767*, 2022. 1
- [6] Shiyu Zhao, Long Zhao, Zhixing Zhang, Enyu Zhou, and Dimitris Metaxas. Global matching with overlapping attention for optical flow estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17592–17601, 2022. 2

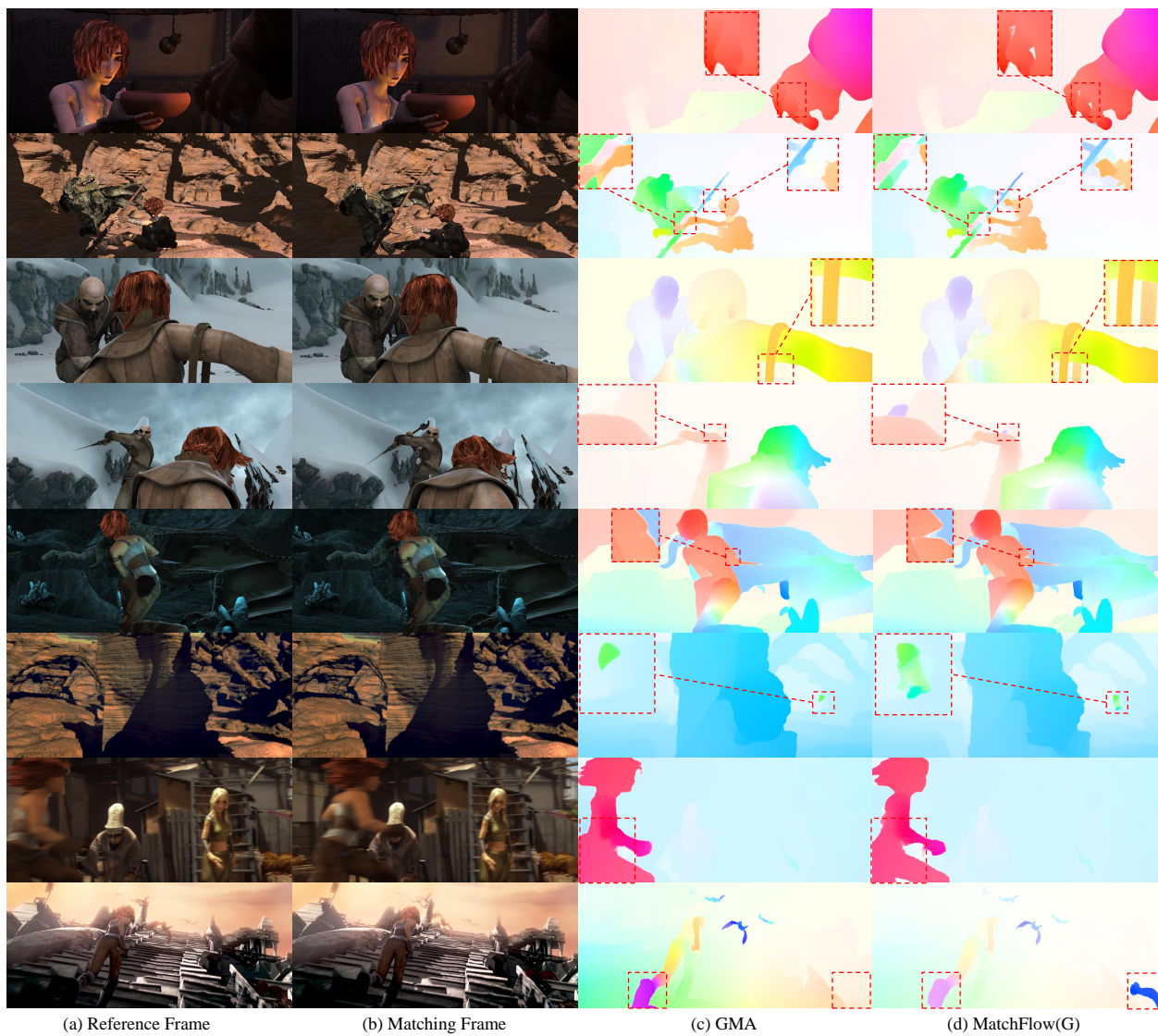


Figure 2. More qualitative results on Sintel test set. First four rows are from clean pass, and the last four from final pass. Ground-truth optical flows are not available and are not shown. Red dashed boxes mark the regions of substantial improvements. Please zoom in for details.

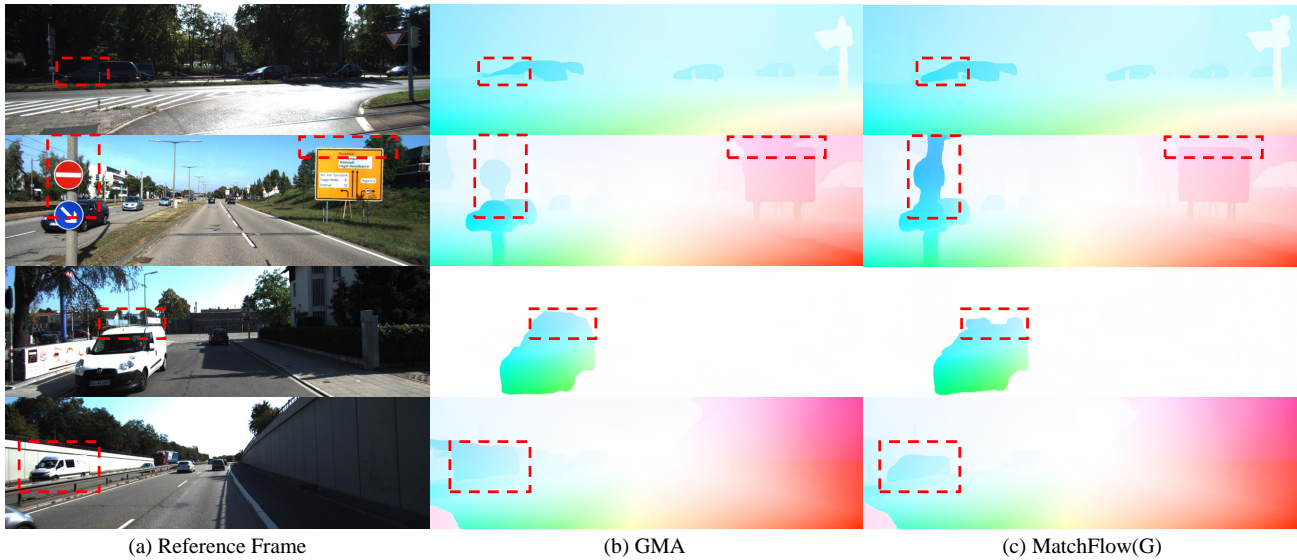


Figure 3. More qualitative results on KITTI test set. Red dashed boxes mark the regions of substantial improvements. Please zoom in for details.

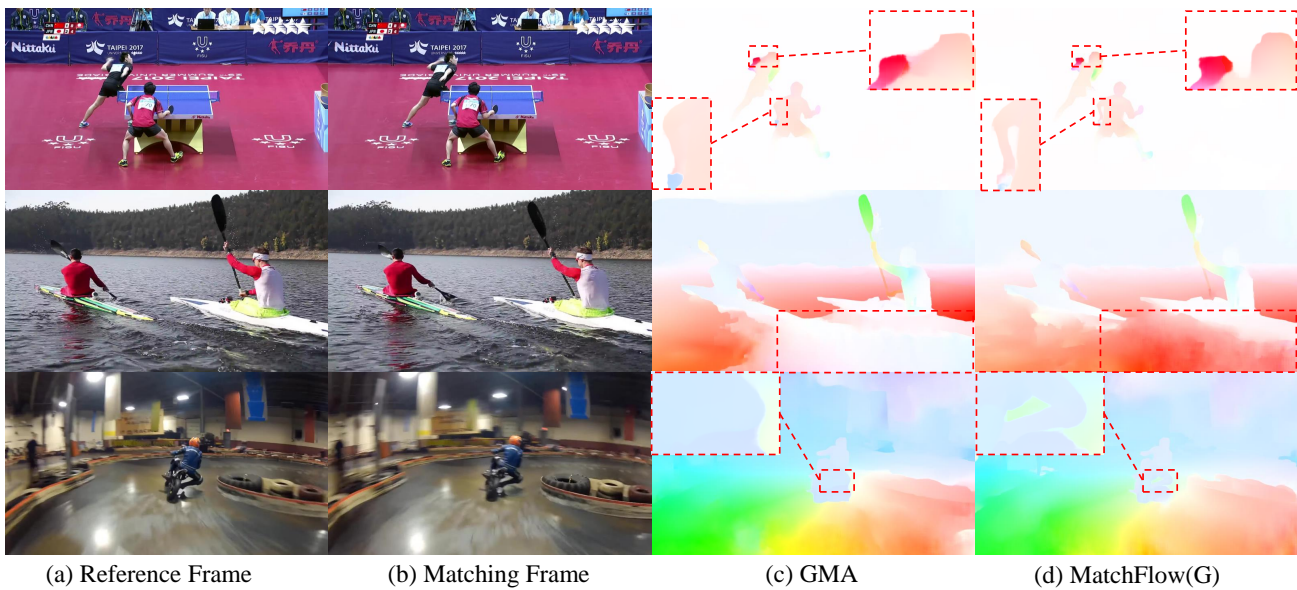


Figure 4. Qualitative results on 1080p (1088x1920) DAVIS [1] test set. Red dashed boxes mark the regions of substantial improvements. Please zoom in for details.

Final Clean

	EPE all	EPE matched	EPE unmatched	d0-10	d10-60	d60-140	s0-10	s10-40	s40+	
GroundTruth ^[1]	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	Visualize Results
GMFlow+ ^[2]	1.028	0.335	6.680	0.868	0.264	0.183	0.227	0.689	5.826	Visualize Results
GMFlow_RVC ^[3]	1.055	0.420	6.227	1.084	0.326	0.227	0.302	0.754	5.513	Visualize Results
FlowFormer++ ^[4]	1.073	0.390	6.635	1.099	0.296	0.179	0.252	0.796	5.810	Visualize Results
SplatFlow ^[5]	1.119	0.511	6.061	1.410	0.394	0.247	0.272	0.868	5.915	Visualize Results
MatchFlow_GMA ^[6]	1.164	0.431	7.130	1.259	0.311	0.197	0.265	0.845	6.387	Visualize Results
RAFT-it+_RVC ^[7]	1.187	0.441	7.260	1.301	0.338	0.181	0.242	0.834	6.723	Visualize Results
FlowFormer ^[8]	1.196	0.406	7.627	1.137	0.310	0.192	0.253	0.800	6.826	Visualize Results
MS_RAFT+_RVC ^[9]	1.232	0.400	8.021	1.101	0.353	0.142	0.159	0.631	8.020	Visualize Results
SKII ^[10]	1.302	0.532	7.571	1.494	0.422	0.225	0.278	0.931	7.269	Visualize Results
SKFlow ^[11]	1.312	0.567	7.379	1.510	0.453	0.231	0.300	0.969	7.159	Visualize Results
MatchFlow_RAFT ^[12]	1.332	0.466	8.390	1.305	0.357	0.208	0.274	0.881	7.670	Visualize Results
MCPFlow_RVC ^[13]	1.346	0.480	8.405	1.268	0.401	0.218	0.263	0.816	8.004	Visualize Results
CGCV ^[14]	1.351	0.556	7.829	1.477	0.459	0.254	0.295	0.920	7.626	Visualize Results
MS_RAFT ^[15]	1.374	0.479	8.678	1.340	0.379	0.224	0.221	0.767	8.572	Visualize Results
SwinTR-RAFT ^[16]	1.379	0.529	8.304	1.272	0.430	0.277	0.325	0.917	7.726	Visualize Results
GMA ^[17]	1.388	0.582	7.963	1.537	0.461	0.278	0.331	0.963	7.662	Visualize Results

(a) Screenshot for Sintel Clean results

Final Clean

	EPE all	EPE matched	EPE unmatched	d0-10	d10-60	d60-140	s0-10	s10-40	s40+	
GroundTruth ^[1]	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	Visualize Results
FlowFormer++ ^[2]	1.943	0.878	10.627	2.302	0.720	0.384	0.438	1.404	10.712	Visualize Results
SplatFlow ^[3]	2.072	1.063	10.285	2.717	0.851	0.452	0.508	1.538	11.109	Visualize Results
FlowFormer ^[4]	2.120	0.986	11.368	2.474	0.791	0.453	0.519	1.470	11.638	Visualize Results
SKII ^[5]	2.160	1.049	11.215	2.754	0.886	0.458	0.465	1.624	11.856	Visualize Results
GMFlow_RVC ^[6]	2.218	1.085	11.437	2.327	0.813	0.651	0.539	1.433	12.428	Visualize Results
SKFlow ^[7]	2.241	1.128	11.314	2.735	0.880	0.507	0.564	1.625	12.048	Visualize Results
MCPFlow_RVC ^[8]	2.350	1.094	12.595	2.702	0.870	0.507	0.517	1.552	13.373	Visualize Results
GMFlow+ ^[9]	2.367	1.095	12.739	2.097	0.808	0.708	0.453	1.328	14.377	Visualize Results
MatchFlow_GMA ^[10]	2.373	1.061	13.070	2.604	0.863	0.485	0.504	1.508	13.735	Visualize Results
CRAFT ^[11]	2.417	1.163	12.637	2.837	1.012	0.547	0.538	1.623	13.656	Visualize Results
CGCV ^[12]	2.430	1.149	12.881	2.821	1.014	0.525	0.500	1.657	13.873	Visualize Results
GMA-FS ^[13]	2.441	1.203	12.551	2.777	0.961	0.594	0.587	1.646	13.576	Visualize Results
ErrorMatch-GMA ^[14]	2.461	1.228	12.519	2.799	1.047	0.642	0.541	1.701	13.821	Visualize Results
AGFlow ^[15]	2.469	1.221	12.643	2.892	0.991	0.698	0.560	1.692	13.816	Visualize Results
GMA ^[16]	2.470	1.241	12.501	2.863	1.057	0.653	0.566	1.817	13.492	Visualize Results
RAFT-OCTC ^[17]	2.574	1.243	13.435	2.880	1.045	0.667	0.578	1.701	14.594	Visualize Results

(b) Screenshot for Sintel Final results

Figure 5. Screenshots for Sintel Clean and Final results on the test server.



22	SwinTR-RAFT	code	4.32 %	6.05 %	4.61 %	100.00 %	0.6 s	8 cores @ 2.5 Ghz (Python)	<input type="checkbox"/>
23	MatchFlow(G)		4.33 %	6.11 %	4.63 %	100.00 %	0.3 s	GPU (Python)	<input type="checkbox"/>
24	RealFlow		4.20 %	6.76 %	4.63 %	100.00 %	0.2 s	8 cores @ 2.5 Ghz (Python)	<input type="checkbox"/>
25	DGA-Flow		4.34 %	6.11 %	4.64 %	100.00 %	0.2 s	1 core @ 2.5 Ghz (Python)	<input type="checkbox"/>
26	FCTR-m		4.45 %	5.63 %	4.65 %	100.00 %	0.2 s	GPU @ 2.5 Ghz (Python)	<input type="checkbox"/>
27	FlowNAS-RAFT-K		4.36 %	6.25 %	4.67 %	100.00 %	0.19 s	GPU @ 2.5 Ghz (Python)	<input type="checkbox"/>
28	FlowFormer	code	4.37 %	6.18 %	4.68 %	100.00 %	0.3 s	GPU (Python)	<input type="checkbox"/>
Z. Huang, X. Shi, C. Zhang, Q. Wang, K. Cheung, H. Qin, J. Dai and H. Li: FlowFormer: A Transformer Architecture for Optical Flow . European conference on computer vision 2022.									
29	CRAFT-intramodes2	code	4.35 %	6.35 %	4.68 %	100.00 %	0.2 s	1 core @ 2.5 Ghz (Python)	<input type="checkbox"/>
30	TPCV+RAFT		4.53 %	5.52 %	4.69 %	100.00 %	0.2 s	1 core 2.5ghz gpu	<input type="checkbox"/>
31	MatchFlow(R)		4.51 %	5.78 %	4.72 %	100.00 %	0.26 s	GPU (Python)	<input type="checkbox"/>
32	UberATG-DRISE		3.59 %	10.40 %	4.73 %	100.00 %	0.75 s	CPU+GPU @ 2.5 Ghz (Python)	<input type="checkbox"/>
W. Ma, S. Wang, R. Hu, Y. Xiong and R. Urtaşun: Deep Rigid Instance Scene Flow . CVPR 2019.									
33	SKII		4.57 %	5.66 %	4.75 %	100.00 %	0.3 s	1 core @ 2.5 Ghz (Python)	<input type="checkbox"/>
34	ErrorMatch-RAFT	code	4.46 %	6.23 %	4.75 %	100.00 %	0.2 s	1 core @ 2.5 Ghz (Python)	<input type="checkbox"/>
35	ErrorMatch-GMA	code	4.53 %	5.87 %	4.75 %	100.00 %	0.3 s	1 core @ 2.5 Ghz (C/C++)	<input type="checkbox"/>
36	Super		4.43 %	6.43 %	4.76 %	100.00 %	0.07 s	GPU @ 2.5 Ghz (Python)	<input type="checkbox"/>
37	RAFT-A	code	4.54 %	5.99 %	4.78 %	100.00 %	0.7 s	GPU @ 2.5 Ghz (Python + C/C++)	<input type="checkbox"/>
D. Sun, D. Vlasic, C. Herrmann, V. Jampani, M. Krainin, H. Chang, R. Zabih, W. Freeman and C. Liu: AutoFlow: Learning a Better Training Set for Optical Flow . CVPR 2021.									
38	CRAFT	code	4.58 %	5.85 %	4.79 %	100.00 %	0.2 s	GPU @ 2.5 Ghz (Python)	<input type="checkbox"/>
X. Sui, S. Li, X. Geng, Y. Wu, X. Xu, Y. Liu, R. Goh and H. Zhu: CRAFT: Cross-Attentional Flow Transformers for Robust Optical Flow . CVPR 2022.									
39	GMFlowNet	code	4.39 %	6.84 %	4.79 %	100.00 %	0.5 s	GPU @ 2.5 Ghz (Python)	<input type="checkbox"/>
S. Zhao, L. Zhao, Z. Zhang, E. Zhou and D. Metaxas: Global Matching with Overlapping Attention for Optical Flow Estimation . CVPR 2022.									

Figure 6. Screenshots for KITTI optical flow evaluation 2015 results on the test server.